

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Кафедра автоматизованих систем обробки інформації та управління

До захисту допущено:

В.о. завідувача кафедри

(підпис) Олександр ПАВЛОВ
(вл.ім'я, прізвище)

“ ____ ” _____ 2020 р.

Дипломний проєкт
на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Інформаційні управляючі
системи та технології»
спеціальності 122 «Комп'ютерні науки та інформаційні технології»**

на тему: «Система з підтримки роботи з клієнтами ресторану »

Виконав:

студент IV курсу, групи ІС-63

Яворський Антон Сергійович

(прізвище, ім'я, по батькові)

(підпис)

Керівник

доц., к.т.н., доц. Жданова Олена Григорівна

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

**Консультант з
графічної
документації**

ст. вик. Проскура Світлана Леонідівна

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

Рецензент

проф. д.т.н. доц. Клименко Ірина Анатоліївна

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент (-ка) _____
(підпис)

Київ – 2020 року

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет (інститут) інформатики та обчислювальної техніки
(повна назва)

Кафедра автоматизованих систем обробки інформації та управління
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп'ютерні науки та інформаційні технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

Олександр ПАВЛОВ
(підпис) (вл.ім'я, прізвище)

“ ” 2020 р.

**ЗАВДАННЯ
на дипломний проєкт студенту**

Яворському Антону Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема проєкту «Система з підтримки роботи з клієнтами ресторану»

керівник проєкту Жданова Олена Григорівна, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “7” травня 2020 р. №1081-с

2. Термін подання студентом проєкту “01” червня 2020 року

3. Вихідні дані до проєкту

Технічне завдання

4. Зміст пояснювальної записки

1. Загальні положення: основні визначення та терміни, опис предметного середовища, огляд ринку програмних продуктів, постановка задачі

2. Інформаційне забезпечення: вхідні дані, вихідні дані, опис структури бази даних

3. Математичне забезпечення: змістовна та математична постановки задачі, обґрунтування та опис методу розв'язання

4. Програмне та технічне забезпечення: засоби розробки, вимоги до технічного забезпечення, архітектура програмного забезпечення, побудова звітів

5. Технологічний розділ: керівництво користувача, методика випробувань програмного продукту

5. Перелік графічного матеріалу

1. *Схема структурна варіантів використання*

2. *Схема структурна станів системи*

3. *Схема бази даних*

4. *Схема структурна класів програмного забезпечення*

5. *Креслення вигляду екранних форм*

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «13» квітня 2020 року

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	<i>Вивчення рекомендованої літератури</i>	<i>07.04.2020</i>	
2.	<i>Аналіз існуючих методів розв'язання задачі</i>	<i>10.04.2020</i>	
3.	<i>Постановка та формалізація задачі</i>	<i>13.04.2020</i>	
4.	<i>Розробка інформаційного забезпечення</i>	<i>20.04.2020</i>	
5.	<i>Алгоритмізація задачі</i>	<i>23.04.2020</i>	
6.	<i>Обґрунтування використовуваних технічних засобів</i>	<i>28.04.2020</i>	
7.	<i>Розробка програмного забезпечення</i>	<i>02.05.2020</i>	
8.	<i>Налагодження програми</i>	<i>06.06.2020</i>	
9.	<i>Виконання графічних документів</i>	<i>07.05.2020</i>	
10.	<i>Оформлення пояснювальної записки</i>	<i>10.05.2020</i>	
11.	<i>Подання ДП на попередній захист</i>	<i>15.05.2020</i>	
12.	<i>Подання ДП на основний захист</i>	<i>01.06.2020</i>	
13.	<i>Подання ДП рецензенту</i>	<i>02.06.2020</i>	

Студент

Антон ЯВОРСЬКИЙ

Керівник

Олена

ЖДАНОВА

[illegible]

Пояснювальна записка до дипломного проєкту

на тему: Система з підтримки роботи з клієнтами ресторану

Київ – 2020 року

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка дипломного проекту складається з п'яти розділів, містить 10 рисунків, 7 таблиць, 1 додаток, 17 джерел.

Дипломний проект присвячений розробці системі для підтримки роботи з клієнтами ресторану.

Задачами дипломного проектування є:

- проведення аналізу вже відомих рішень рекомендаційних систем;
- розробка способу надання рекомендацій щодо вибору страв;
- облік та аналіз анкетних даних користувачів;
- облік та аналіз оцінок роботи рекомендаційної системи від користувачів;
- розробка інтерфейсу рекомендаційної системи.

Ціль автоматизованої системи – підтримка клієнтів ресторану шляхом надання рекомендацій, враховуючи індивідуальні переваги.

У розділі інформаційного забезпечення було здійснено опис предметного середовища, процесу діяльності, функціональної моделі, а також наведено детальний опис наявних аналогів з зазначенням їхніх переваг та недоліків.

Розділ математичного забезпечення присвячений обґрунтуванню вибору методу розв'язку поставленої задачі з попереднім розглядом різних можливих підходи для надання рекомендацій. Було запропоновано метод, який будує дерево прийняття рішень, використовуючи алгоритм ID3 та C4.5 для надання рекомендацій щодо вибору страв, опираючись на переваги клієнта.

					ДП 6331.00.000.ПЗ				
Зм.	Арк.	Прізвище	Підпис	Дата					
Розроб.		Яворський А.С.			Система з підтримки роботи з клієнтами ресторану	Літ.		Арк.	Аркушів
Перевірив.		Жданова О.Г.						1	
						КПІ ім. Ігоря Сікорського Каф. АСОІУ гр. ІС-63			
Н. кон.		Проскура С.Л.							
Затв.		Павлов О.А.							

У розділі програмного та технологічного забезпечення наведено детальний опис засобів розробки, вимог до технічного забезпечення, а саме вимоги до клієнтських робочих станцій, серверу та серверу СУБД.

Технологічний розділ присвячений керівництву користувача та проведенню випробувань над створеним програмним продуктом.

РЕКОМЕНДАЦІЙНА СИСТЕМА, РЕКОМЕНДАЦІЯ, КЛІЄНТ, СТРАВА, ДЕРЕВО ПРИЙНЯТТЯ РІШЕНЬ, АЛГОРИТМ ІДЗ, АЛГОРИТМ С4.5.

					ДП 6331.00.000 ПЗ	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

ABSTRACT

The structure and scope of work. Explanatory note diploma consists of five chapters that contain 10 pictures, 7 tables, 1 application, 17 sources.

Diploma project is about developing a client-support system that will base its decisions on the client preferences.

The goals of automated system:

- analysis of known solutions of recommendation systems;
- developing a method for producing recommendations for choosing a meal;
- accounting and analysis statement users;
- accounting estimates and analysis of the recommendation system by the users;
- creating an interface of the recommendation system.

The purpose of the automated system to produce a list of meal recommendations and to simplify user's selection of meals during their visits to the restaurant.

The information part contains a description of all the processes, functional model, as well as the detailed description of the somewhat similar products, indicating their advantages and disadvantages.

The mathematical support part is about choosing a method for solving the problem with the consideration of various different paths for making recommendations. The decision-making tree has been built by using the ID3 and C4.5 algorithm.

					ДП 6331.00.000 ПЗ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

The software support part contains a description of software tools, architecture and system requirements such as the requirements for client workstations or database server.

The chapter of engineering support contents user manual and some of the test methods.

RECOMMENDATION SYSTEM, RECOMMENDATIONS, CLIENT, DECISION TREE, ALGORITHM ID3, C4.5 ALGORITHM.

					ДП 6331.00.000 ПЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

Пояснювальна записка до дипломного проєкту

на тему: Система з підтримки роботи з клієнтами ресторану

Київ – 2020 року

ЗМІСТ

<u>ВСТУП</u>	8
<u>1. ЗАГАЛЬНІ ПОЛОЖЕННЯ</u>	10
<u>1.1 Опис предметного середовища</u>	10
<u>1.1.1 Опис процесу діяльності</u>	10
<u>1.2 Огляд наявних аналогів</u>	13
<u>1.3 Постановка задачі</u>	14
<u>1.3.1 Призначення розробки</u>	14
<u>1.3.2 Цілі та задачі розробки</u>	14
<u>Висновок до розділу</u>	14
<u>2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ</u>	15
<u>2.1 Вхідні дані</u>	15
<u>2.2 Вихідні дані</u>	16
<u>2.3 Опис структури бази даних</u>	17
<u>Висновок до розділу</u>	18
<u>3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ</u>	19
<u>3.1 Змістовна постановка задачі</u>	19
<u>3.2 Математична постановка задачі</u>	19
<u>3.3 Обґрунтування методу розв'язання</u>	20
<u>3.3.1 Алгоритм ID3</u>	21
<u>3.3.2 Алгоритм C4.5</u>	22
<u>3.3.4 Опис методу розв'язання</u>	23
<u>Висновок до розділу</u>	24
<u>4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ</u>	25

<u>4.1 Засоби розробки</u>	25
<u>4.1.1 Мова програмування С#</u>	25
<u>4.1.2 Система керування базами даних MySQL</u>	25
<u>4.2 Вимоги до технічного забезпечення</u>	25
<u>4.2.1 Вимоги до клієнтських робочих станцій</u>	28
<u>4.2.2 Вимоги серверу СУБД</u>	29
<u>4.3 Архітектура програмного забезпечення</u>	29
<u>4.3.1.1 Діаграма класів</u>	29
<u>4.3.2 Діаграма послідовності</u>	30
<u>4.3.3 Діаграма компонентів</u>	31
<u>Висновки до розділу</u>	32
<u>5. ТЕХНОЛОГІЧНИЙ РОЗДІЛ</u>	33
<u>5.1 Керівництво користувача</u>	33
<u>5.2 Випробування програмного продукту</u>	34
<u>5.2.1 Мета випробувань</u>	35
<u>5.2.2 Загальні положення</u>	35
<u>5.2.3 Результати випробувань</u>	35
<u>Висновок до розділу</u>	37
<u>ЗАГАЛЬНІ ВИСНОВКИ</u>	38
<u>ПЕРЕЛІК ПОСИЛАНЬ</u>	40

ВСТУП

Багато людей при першому потраплянні в певний ресторан часто губляться, коли необхідно обрати страву, замовити їжу для декількох людей, для дітей, тощо. Деякі ресторани пропонують так звані «Страви дня» чи «Пропозиція шеф-повара», проте вони не орієнтуються на бажання та переваги клієнта, а, здебільшого, лише на кількість продуктів, зроблених замовлень. Для того, щоб допомогти клієнтам ресторану при виборі замовлення, можна запропонувати систему, яка б за певними критеріями та знанням про переваги людини, сама визначала, що саме тому чи іншому клієнту можна рекомендувати. Таку систему будемо називати рекомендаційною.

Рекомендації - це певні інструкції певній людині щодо вибору того чи іншого варіанту. Рекомендації також є особливою формою фільтрації інформації, що використовує попередні поведінки користувачів та подібність для створення списку інформації, що враховує особисті переваги кінцевих користувачів.

Є багато різновидів рекомендаційних систем, що намагаються дізнатись та передбачити, що саме буде цікаве покупцю чи клієнту і це реально зробити, знаючи певні дані про їх вподобання.

Система рекомендацій - це активне поле досліджень у галузі видобутку даних та машинного навчання. До них відносять застосунки, що намагаються передбачити такі об'єкти, які були б цікаві користувачеві, враховуючи певну інформацію, надану клієнтом.

Основними характеристиками рекомендаційних систем є:

- Предмет рекомендації (опис об'єкта рекомендації);

- мета рекомендації (причини рекомендації, наприклад, здійснення замовлення, здійснення покупок, тощо);
- контекст рекомендації (чим зайнятий користувач під час формування рекомендацій);
- джерело рекомендації (на чому або на якому досвіді основані рекомендації);
- ступінь персоналізації (варіанти рекомендацій, в залежності від того, чи рекомендують певний об'єкт всім, чи це є лише персональною рекомендацією, чи використовуються надані дані клієнта, тощо);
- прозорість (обґрунтування рекомендацій);
- формат рекомендації (інтерфейс та доступність застосунку);
- алгоритми (алгоритми, що використовуються для збору інформації та формування рекомендацій).

Дипломний проект присвячений розробці системи підтримки клієнтів ресторану.

Метою системи є мінімізація зусиль для оформлення замовлення шляхом використання автоматично створених рекомендацій.

Практичне значення одержаних результатів. Розроблено систему для підтримки клієнтів ресторану шляхом надання рекомендацій щодо вибору страв з урахуванням переваг.

					ДП 6331.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

1. Загальні положення

1.1 Опис предметного середовища

Клієнти ресторану часто зіштовхуються з багатьма проблемами при отриманні рекомендацій в різних закладів харчування. Ці проблеми варіюються від нерозуміння чи загальної відсутності користувацького інтерфейсу для правильного сприйняття інформації, генералізація рекомендацій по вибору меню та інші.

Дана система буде здійснювати власне підтримку клієнтів ресторану шляхом аналізу та детальної обробки наданих клієнтом переваг, створення та надання переваг, взявши за основу меню наявних в ресторані страв. Ці переваги можуть бути виражені в різних формах, тому було вирішено розділити їх на різні таблиці для зручнішої роботи з даними.

1.1.1 Опис процесу діяльності

Коли клієнт вперше заходить до ресторану, працівник ресторану надає йому анкету або усно дізнається про його переваги. Після цього дані заносяться в систему, що власне й надає рекомендації щодо вибору страви.

Адміністратор системи здійснює періодичні оновлення та виправлення неточностей даних, занесених в систему.

Менеджер ресторану слідкує за виконанням всіх процесів, відповідає та регулює роботу між персоналом ресторану та системою.

Наведемо структурну схему станів системи на рисунку 1.1.

					ДП 6331.00.000 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

Початок діяльності

Вхід в систему

Анкета була раніше
заповнена?

Ні

Так

Ввести
користувацькі дані

Заповнити анкету

Отримати
рекомендації

Завершення діяльності

Рисунок 1.1 – Схема структурна станів системи

1.1.2 Опис функціональної моделі

В даній системі є 2 актори, а саме: клієнт та адміністратор. Нижче на рисунку 1.2 наведена діаграма варіантів використання системи, що відображає головні процеси автоматизації.

Змн.	Арк.	№ докум.	Підпис	Дата



Рисунок 1.2 – Діаграма варіантів використання

При вході в систему клієнт здійснює одну з двох дій:

- Авторизується в системі, використовуючи записані в базу дані.
- Переходить на екран заповнення анкети де власне і надає дані, необхідні для створення рекомендацій, що б підходили під його переваги та смаки.

Після цього надані дані надсилаються в систему і здійснюється їх обробка. В ході роботи програмного забезпечення здійснюється побудова дерева прийняття рішень і на його основі формуються рекомендації. Детальніше про сам процес побудови дерева прийняття рішень наведено в розділі про математичне забезпечення.

Функціональні вимоги системи наведені та таблиці 1.1.

Таблиця 1.1 Функціональні вимоги системи

Функціональна вимога	Пріоритет
1. Система має здійснювати ведення страв.	Високий
2. Система має здійснювати ведення типів замовлень.	Високий
3. Система має здійснювати ведення питань щодо побажання клієнтів.	Високий
4. Система повинна формувати рекомендації.	Високий
5. Система повинна формувати звітності.	Середній

1.2 Огляд наявних аналогів

В ході процесу пошуку аналогічних рішень, було виявлено декілька програмних продуктів, що намагались вирішити аналогічні або подібну задачу. Однак для оптимізації рішення, вирішено було поділити задачу на декілька підзадач. Порівняння існуючих аналогів наведено в таблиці 1.1.

Таблиця 1.1 – Порівняння існуючих аналогів системи підтримки клієнтів ресторану

Назва системи	Функціонал
«РестАрт»	Автоматизація робочих місць працівників закладу харчування, налаштування та управління цими робочими місцями. Створення рекомендацій для вибору страв клієнтами шляхом оцінки продажу певних страв.
«ULTRA» Ресторан	Розмежування повноваження працівників, структурування бази даних клієнтів, внесення правок в меню, тарифікація часу перебування клієнта.
Програмний комплекс	Оптимізація бізнес-процесів, контроль роботи персоналу, підвищення якості обслуговування клієнтів, зменшення

«jSolutions»

часу обслуговування клієнтів, контроль роботою мережі
дистанційно.

1.3 Постановка задачі

1.3.1 Призначення розробки

Система для підтримки клієнтів ресторану призначена для допомоги клієнтам ресторану при виборі замовлення, створення рекомендацій, враховуючи при цьому всі переваги клієнтів.

1.3.2 Цілі та задачі розробки

Метою створення системи є мінімізація зусиль для оформлення замовлення шляхом використання автоматично створених рекомендацій.

Для реалізації поставленої мети, система має виконувати наступні функції:

- функція ведення страв;
- функція ведення типів замовлення;
- функція ведення питань щодо побажання клієнтів
- функція формування рекомендацій;
- функція формування звітності.

Висновок до розділу

На даному етапі розробки дипломного проекту було проведено аналіз предметного середовища, описано мету, призначення та функції майбутнього програмного продукту.

2 Інформаційне забезпечення

2.1 Вхідні дані

Для створення програмного забезпечення було проведено детальний аналіз функціоналу, на основі якого було обрано технології розробки. Для розробки програмного продукту було обрано C# .NET, а для роботи з базами даних використовувався інструментарій MySQL.

До вхідних даних системи належать логін та пароль користувача або ж анкета, в якій клієнт задає всі необхідні для роботи програми дані при першому відвідуванні ресторану.

При запуску програмного забезпечення, клієнт бачить наступну екранну форму головного меню(Рисунок 2.1).

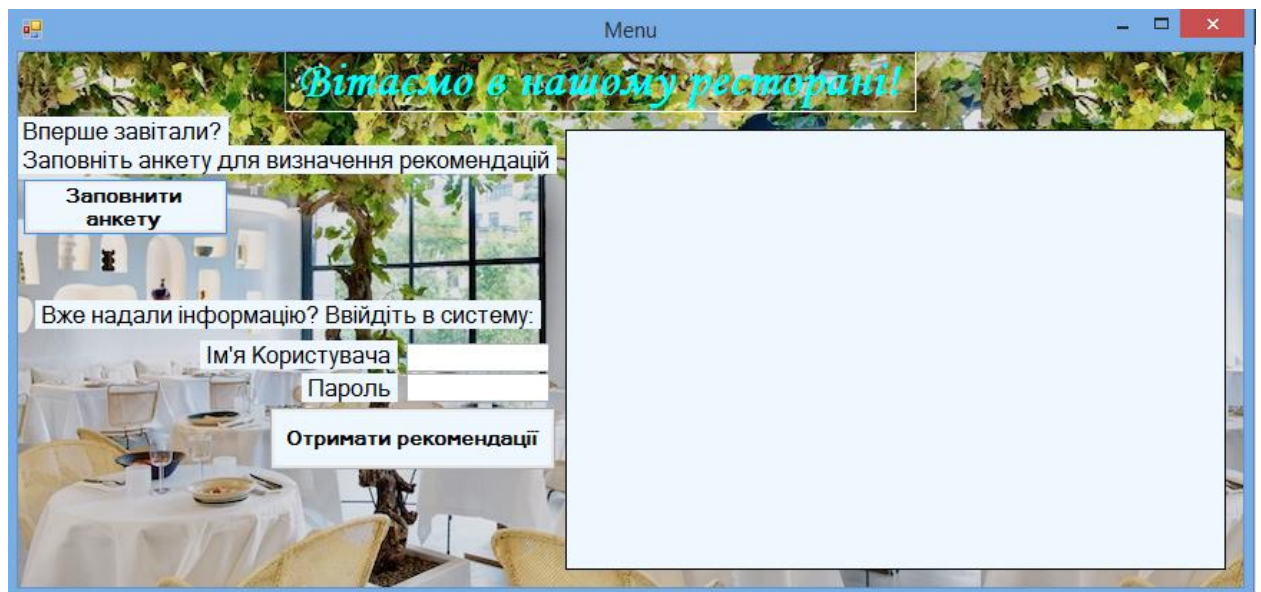


Рисунок 2.1 – Екранна форма головного меню програми

Якщо користувач вже був раніше зареєстрований в базі даних цього ресторану, то після коректного вводу особистих даних треба натиснути на кнопку «Отримати рекомендації».

Якщо ж користувач вперше в ресторані, то для визначення його переваг при виборі страв, йому необхідно буде заповнити анкету. Екранна форма користувача подана на рисунку 2.2.

The screenshot shows a web form titled "MENU" within a window labeled "User". The form is designed with a rustic theme, featuring a wooden background and images of a wooden fork and spoon. The form fields include:

- Ім'я** (Name): Text input field.
- Прізвище** (Surname): Text input field.
- Пароль** (Password): Text input field.
- Повторити пароль** (Repeat password): Text input field.
- Вік** (Age): Text input field.
- Цінова категорія** (Price category): Dropdown menu.
- Переваги в виборі їжі** (Food preferences): Checkboxes for:
 - М'ясна (Meat)
 - Рослинного походжу (Plant-based)
 - Веганська (Vegan)
 - Супи (Soups)
 - Салати (Salads)
 - Крупи (Grains)
- Напої** (Drinks): Checkboxes for:
 - Алкогольні (Alcoholic)
 - Соки (Juices)
 - Газована вода (Carbonated water)
 - Чай (Tea)
- Улюблена страва** (Favorite dish): Text input field.
- Відправити анкету** (Submit survey): Red button at the bottom.

Рисунок 2.2 – Екранна форма меню клієнта

Після вводу всіх необхідних полів, система записує надані дані в базу та здійснює подальшу їх обробку.

2.2 Вихідні дані

Вихідними даними системи є сформовані, індивідуальні для кожного клієнта, рекомендації щодо вибору страв, згідно з їх перевагами, основуючись на вхідних даних. Всі ці дані виводяться в таблицю екранної форми головного меню програми.

2.3 Опис структури бази даних

Наведемо список таблиць та полів бази даних.

№	Назва таблиці	Поля	Тип даних
1	Страви	Id страви Назва страви Складові	Integer String String
2	Замовлення	Id замовлення Номер страви Назва страви Загальна вартість замовлення	Integer Integer String Integer
3	Анкета	Id клієнта Користувачьке ім'я клієнта Пароль Наявність алергій Складові страв Цінова категорія Вид меню	Integer String String Boolean String array Integer String
4	Рекомендація	Id рекомендації Номер рекомендованої страви Назва рекомендованої страви Короткий опис рекомендації	Integer Integer String String

5	Меню	Назва меню Id страви Назва страви Складові страви Ціна	String Integer String String Integer

Висновок до розділу

У даному розділі було вказано необхідні вхідні та отримані вихідні дані, а також було виконане проектування власне бази даних, описані всі таблиці та поля.

3 Математичне забезпечення

3.1 Змістовна постановка задачі

Наразі існує багато відомих ресторанів, в тому числі і з онлайн-замовленнями та різними функціями, пропонованими клієнтам. Проте не всі ресторани враховують смаки та переваги своїх відвідувачів. А навіть ті, що намагаються вирішити цю проблему, здебільшого орієнтуються на кількість продаж того чи іншого продукту чи страви.

Цей підхід не завжди є вірним, тому часто клієнти є незадоволеними по причині того, що обрали страву не вірно чи просто не змогли оцінити її по наданому меню. Через це виникла необхідність створити автоматизовану систему для надання рекомендацій, опираючись на смаки та переваги відвідувачів.

Проте навіть у цього рішення є певні проблеми, а саме неможливість визначення того, що саме необхідно порадити, без залучення або мінімізації залучення до цього процесу клієнта. Доречним вирішенням проблеми було б анкетування всіх нових клієнтів при першому заході до ресторану. Це дозволило б системі мати певні початкові знання про переваги та вподобання відвідувачів та створення унікальних рекомендацій. Ця проблема також вирішується знаходженням подібних за наданням переваг клієнтів. Але для цього необхідно буде оновлювати дерево прийняття рішень після отримання додаткової інформації.

3.2 Математична постановка задачі

Нехай маємо певну множину рядків даних. Кожна пара (властивість, значення), що має таку ж структуру задає вектор. Кожна з властивостей задає

векторну категорію. Так як поставленим завданням є створення такого дерева прийняття рішень, що ґрунтуючись на певних властивостях, які не відповідають безпосередньо за категорію, необхідно дослідити значення категоризаційної властивості. В більшості випадків це можна задати як булевий тип даних, що приймає лише 2 значення, - «вірно»(true), або хибне(false), але завжди одне із значень буде означати невдачу. Враховуючи вищесказане, оберемо певну множину входів як $A = \{(a_i, p_i)\}$, де a_i є атрибутом, а p_i – це фактор впевненості для атрибуту a_i . Звідси $B = (m_i, q_i)$, де m_i – це об’єкт, що рекомендується, а q_i буде визначати степінь приналежності об’єкту, що рекомендується. Задля отримання результату необхідно побудувати $f: A \rightarrow B$, так, щоб найбільше враховувались саме переваги клієнта.

3.3 Обґрунтування методу розв’язання

Для вирішення даної задачі було обрано алгоритми, що будують дерева прийняття рішень.

Дерево прийняття рішень - структура дерева, схожа на блок-схему, де внутрішній вузол представляє особливість (або атрибут), гілка представляє правило рішення, а кожен вузол листів представляє результат. Найвищий вузол у дереві рішень відомий як кореневий вузол. Він розділяє дерево рекурсивно, що називають рекурсивним поділом. Ця структура, схожа на блок-схему, допомагає у прийнятті рішень. Розглянемо алгоритми побудови дерева прийняття рішень (Таблиця Error! No text of specified style in document..1).

Таблиця Error! No text of specified style in document..1 – Класифікація алгоритмів, які будують дерева

					ДП 6331.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

Назва алгоритму	Опис
ID3	Базовий алгоритм побудови дерева прийняття рішень «розділяй та володарюй»
C4.5	Є розширенням алгоритму ID3.

Необхідно зазначити, що з дерева прийняття рішень можна отримати правила виведення та навпаки.

3.3.1 Алгоритм ID3

Алгоритм ID3 використовується для побудови дерев прийняття рішень, маючи множину некатегоризаційних властивостей C_1, C_2, \dots, C_n , категоризаційну властивість C , і множину записів для навчання T .

Алгоритм ID3 вибирає найкращу функцію на кожному кроці під час створення дерева рішень. ID3 використовує інформаційне посилення або просто посилення, щоб знайти найкращу функцію. Інформаційний приріст обчислює зменшення ентропії та вимірює, наскільки добре дана функція відокремлює або класифікує цільові класи. Функція, що має найбільший приріст інформації, вибирається як найкраща.

Простими словами, ентропія є мірою розладу, а ентропія набору даних - міра розладу в цільовій особливості набору даних.

У разі бінарної класифікації (де у цільовому стовпчику є лише два типи класів) ентропія дорівнює 0, якщо всі значення в цільовому стовпчику є однорідними (подібними) і становитимуть 1, якщо цільовий стовпець має однакові числові значення для обох класів.

Позначимо наш набір даних як S , ентропія обчислюється як:

$$\text{Entropy}(S) = - \sum p_i * \log_2(p_i) ; i = 1 \text{ to } n ,$$

де n - загальна кількість класів у цільовому стовпчику

p_i - ймовірність класу 'i' або відношення «кількості рядків з класом i в цільовому стовпчику» до «загальної кількості рядків» у наборі даних.

Приріст інформації для стовпця A функції обчислюється як:

$$IG(S, A) = Entropy(S) - \sum((|S_v| / |S|) * Entropy(S_v))$$

де S_v - сукупність рядків у S, для яких стовпець A має значення v, $|S_v|$ - кількість рядків у S_v і аналогічно $|S|$ - кількість рядків у S.

Для використання алгоритму ID3 необхідно здійснити такі кроки:

- Обчислити інформаційний приріст кожної функції.
- Враховуючи, що всі рядки не належать до одного класу, розділити набір даних S на підмножини, використовуючи функцію, для якої приріст інформації максимальний.
- Створити вузол дерева рішень за допомогою функції з максимальним збільшенням інформації.
- Якщо всі рядки належать до одного класу, то поточний вузол робимо листом з класами як його мітками.
- Повторити для інших функцій, поки не вичерпаються всі функції, або у дереві рішень не закінчатся листи.

3.3.2 Алгоритм C4.5

C4.5 є доповненням до алгоритму ID3, що враховує допустимі значення, недетерміновані значення, відсікання дерева, виведення правил та інше.

C4.5 - алгоритм, розроблений Россом Квінланом, який генерує дерева рішень (DT), які можна використовувати для задач класифікації. Він вдосконалює (розширює) алгоритм ID3, розглядаючи як безперервні, так і дискретні атрибути, відсутні значення та обрізку дерев після будівництва. Його комерційний наступник - C5.0 / See5, набагато швидше, ніж C4.5,

ефективніше пам'яті та використовується для побудови менших дерев рішень.

Будучи контрольованим алгоритмом навчання, він вимагає набору навчальних прикладів, і кожен приклад може розглядатися як пара: об'єкт введення та бажане вихідне значення (клас). Алгоритм аналізує навчальний набір і будує класифікатор, який повинен вміти правильно класифікувати як навчальні, так і тестові приклади. Тестовий приклад - це об'єкт введення, і алгоритм повинен передбачати вихідне значення (приклад повинен бути призначений класу).

Класифікатор, який використовується в С4.5, - це рішення рішень, і це дерево будується від кореня до листя. Враховуючи два правильних рішення певної проблеми, ми повинні вибрати більш просте рішення.

С4.5 додає цілий ряд доповнень до оригінального алгоритму ID3. Під час побудови дерева рішень ми можемо мати справу із навчальними даними, що мають рядки із невідомими значеннями властивостей під час обрахунку приросту, беручи до уваги лише рядки, де ця властивість визначена. Під час використання дерева, ми можемо класифікувати рядки, що мають невідомі значення властивостей, вгадуючи ймовірності появи різних результатів.

3.3.4 Опис методу розв'язання

В деревах рішень майже завжди вибираються правильні рішення (для навчальної вибірки), так як вони завжди ґрунтуються на навчальних даних. Часто буває так, що саме шлях по дереву буде занадто довгим, тому здійснюється відсікання дерева. Його суть полягає в тому, що ціле дерево

замінюється листом, тобто коли дерево виявляє, що у піддереві очікувана помилка більша ніж в певному листі, відбувається заміна.

Тест Фішера можна використовувати, якщо необхідно визначити чи справді від некатегоричного атрибута залежить вся категорія атрибутів.

Для дерева прийняття рішень створимо правило. Для кожного шляху від дерева до листа будемо записувати правило. Результатом цього буде те, що ліва сторона буде швидко будуватись з наявних міток вузлів і дуг, що з'єднують між собою вершини. Цей набір може також бути спрощеним. Якщо α' отриманий з α шляхом вилучення деяких умов, то α замінюємо за допомогою α' в наявному правилі, а в множині навчання з кожного вектора, що задовольняє і α' і α . Якщо ж більш ніякі правила не можна застосувати, то правило можна видалити.

Висновок до розділу

В даному розділі було визначено змістовну та математичну постановку задачі, розглянуто різні можливі підходи для надання рекомендацій, обґрунтовано вибір методу розв'язку поставленої задачі, а також запропоновано метод, який будує дерево прийняття рішень.

					ДП 6331.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

4 Програмне та технічне забезпечення

4.1 Засоби розробки

Засоби розробки

Для розробки програмного забезпечення дипломного проекту, було використано мову програмування C# та систему для управління баз даними MySQL.

4.1.1 Мова програмування C#

C # - це нова мова, створена корпорацією Майкрософт та подана в ECMA для стандартизації. Ця нова мова була створена командою Microsoft на чолі з Андерсом Хайльсбергом. За допомогою C # вони зосередилися на тому, щоб визначити, що було б правильно щодо існуючих мов, та як саме можна додати вдосконалення, щоб зробити щось вже існуюче кращим.

C # є потужною та доволі гнучкою мовою програмування. Як і всі мови програмування, її можна використовувати для створення різноманітних програм. Потенціал C # обмежений лише уявою користувачів. C # використовується в різних проектах, таких як динамічні веб-сайти, засоби розробки та навіть компілятори.

C # створена як об'єктно-орієнтована мова програмування (ООР). Інші мови програмування включають об'єктно-орієнтовані функції, але мало які з них мають повністю об'єктно-орієнтовані функції.

Багато людей вважають, що нова мова програмування не потрібна, оскільки Java, C ++, Perl, Microsoft Visual Basic та інші існуючі мови надають

					ДП 6331.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

усі необхідні функції, але C # - це мова, що походить від C і C ++, і створена вона з нуля.

Microsoft почала використовувати C і C ++ і представила нові функції, що полегшують використання цих мов. Багато з цих функцій дуже схожі на функції Java. Зрештою, у Microsoft була поставлена низка цілей при створенні мови.

Ці цілі можна узагальнити у твердженнях Microsoft щодо C #:

- C # є простою мовою програмування.
- C # є сучасною мовою програмування.
- C # є об'єктно-орієнтованою мовою програмування

Окрім причин, визначених Microsoft, існують й інші причини використання C #:

- C # є потужною і гнучкою.
- C # є мовою "з кількох слів".
- C # є модульною.

Обробка винятків, збирання сміття, розширювані типи даних та захист коду - це функції, що визначають сучасні мови програмування. C # містить усе це.

Ключовими для об'єктно-орієнтованих мов є інкапсуляція, успадкування та поліморфізм. C # підтримує все це. Інкапсуляція - це розміщення функціональних можливостей в єдиний пакет. Спадкування - це структурований спосіб розширення існуючого коду та функціональності на

нові програми та пакети. Поліморфізм - це здатність адаптуватися до того, що потрібно зробити

4.1.2 Система керування базами даних MySQL

База даних - це фактично структурований збір даних. Нею може бути як просте меню ресторану, так і значна по місткості база даних покупців всієї мережі. Для отримання доступу до редагування даних, необхідна певна система для керування базами даних, чим в даному проєкті виступає MySQL Server. Оскільки комп'ютери не можуть обробляти велику кількість даних без великих затрат ресурсів, система управління базами даних відіграє центральну роль в обчисленні.

MySQL для Visual Studio забезпечує доступ до об'єктів і даних MySQL з Visual Studio. Як пакет Visual Studio, MySQL для Visual Studio інтегрується безпосередньо в провіднику сервера, надаючи можливість створювати нові з'єднання та працювати з об'єктами бази даних MySQL.

Поняття функціональності включає:

Розробка баз даних безпосередньо в MySQL дозволяє інтегрувати безпосередньо в Visual Studio, об'єкти баз даних (таблиці, представлення даних, збережені підпрограми, тригери, індекси тощо), а також створювати, змінювати або видаляти безпосередньо в провіднику сервера.

Редактори візуальних об'єктів містять корисну інформацію, яка допоможе провести процес редагування. Також доступні стандартні представлення даних, щоб допомогти переглянути збережені в базі даних дані.

Інструмент дизайну запитів Visual Studio також підтримується системою для управління базами даних MySQL. За допомогою цього інструменту ви можете запитувати та переглядати дані з таблиць чи

представлень, одночасно комбінуючи фільтри, групові умови та параметри. Збережені підпрограми (як з параметрами, так і без них) також можна запитувати.

Використовуючи стандартне середовище та елементи управління Visual Studio, можна встановлювати точки перерви, додавати таймер, входити в різні режими, виходити з них та виконувати дії над ними. Локальні змінні можуть бути додані до вікна перегляду, а також підтримується навігація по стеку дзвінків.

Entity Framework підтримується, щоб дозволити генерування коду на основі шаблонів та повну підтримку дизайнерів та «wizard»'ів.

MySQL є реляційною системою для керування базами даних, що означає, що база даних зберігає дані в окремих таблицях, а не зберігає всі дані в одному великому сховищі.

4.2 Вимоги до технічного забезпечення

4.2.1 Вимоги до клієнтських робочих станцій

Комп'ютер на базі Intel-сумісного процесору, починаючи з процесора Celeron, з тактовою частотою 700 мегагерц, рекомендовано – Pentium 4 з тактовою частотою 1400 мегагерц та вище;

Об'єм оперативної пам'яті — 256 Мб, рекомендований об'єм — 512 Мб й більше;

Мінімальний об'єм вільного дискового простору — 10 гігабайт, рекомендований об'єм — 40 гігабайт та більше;

Монітор з підтримкою мінімального роздільної здатності екрану 800 на 600 пікселів; для зручної роботи рекомендується роздільна здатність 1024 на 768 пікселів та більше.

					ДП 6331.00.000 ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

Вимоги до програмного забезпечення:

Особливих вимог до програмного забезпечення немає.

Вимоги до комп'ютерної мережі

Особливих вимог до комп'ютерної мережі немає.

4.2.2 Вимоги серверу СУБД

В якості серверу бази даних був вибрана СУБД MySQL не нижче 5 версії. СУБД MySQL менш вимоглива до системних ресурсів та більш швидка, ніж її прямі конкуренти. Також СУБД MySQL є вільно розповсюджуваною, тобто ліцензія на її використання безкоштовна, що дозволяє знизити кінцеву вартість системи.

4.3 Архітектура програмного забезпечення**4.3.1.1 Діаграма класів**

На рисунку 4.1 зображена структурна схема класів програмного забезпечення розробленої програми.

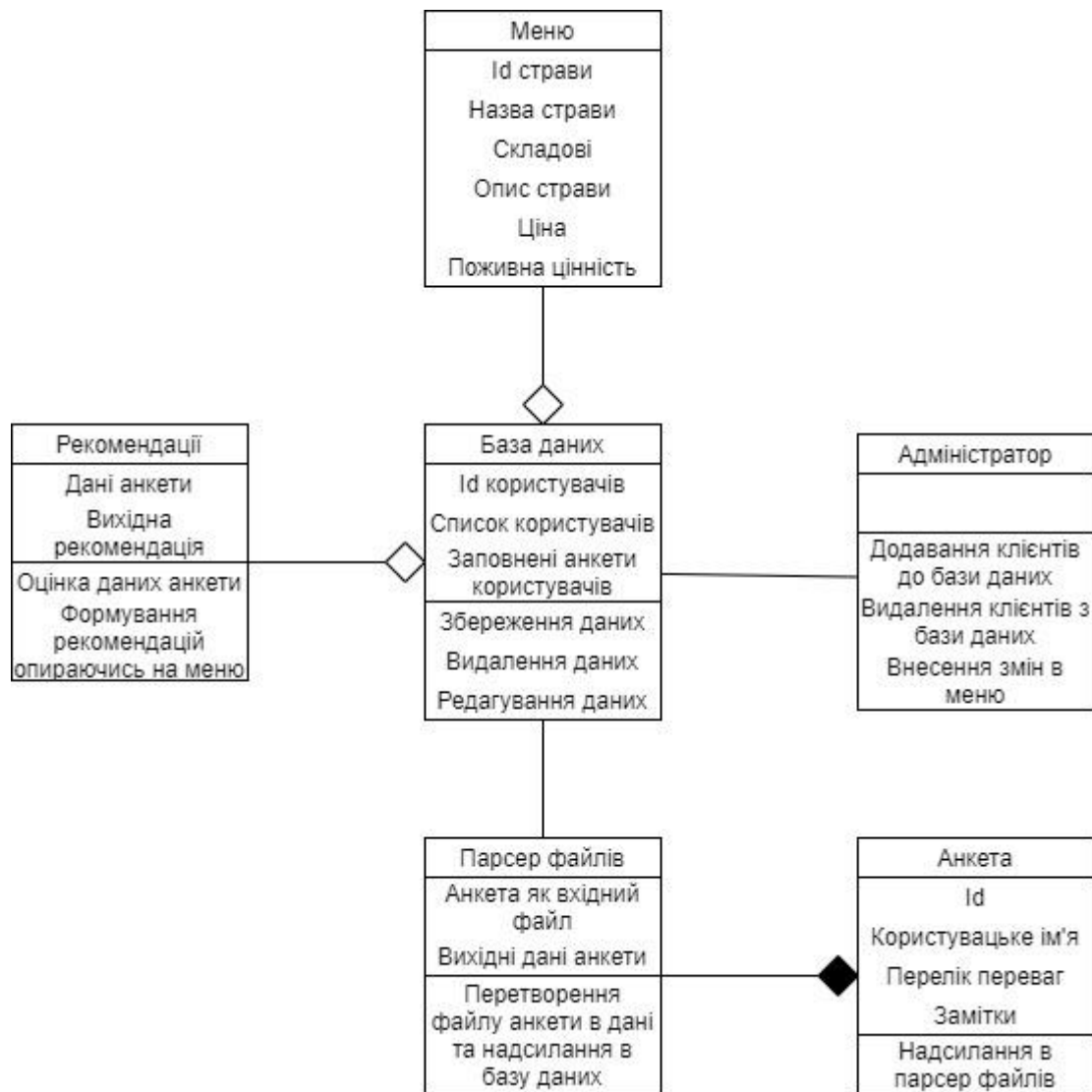


Рисунок 4.1 Схема структурна класів програмного забезпечення

4.3.2 Діаграма послідовності

Для представлення типової послідовності дій, що виконуються при наданні рекомендацій, представимо схему структури послідовності на рисунку 4.2.

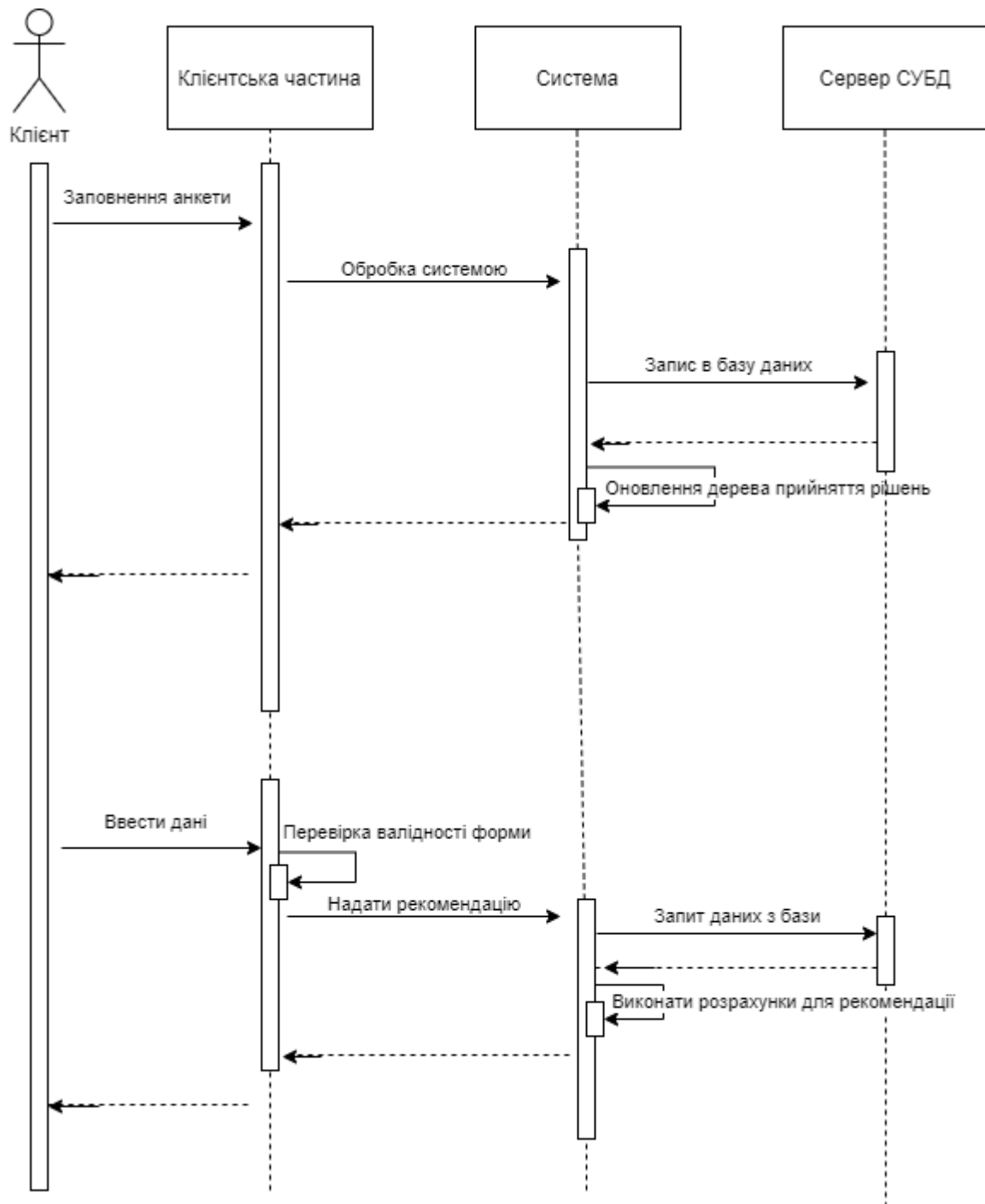


Рисунок 4.2 Схема структурної послідовності

4.3.3 Діаграма компонентів

Змн.	Арк.	№ докум.	Підпис	Дата

На рисунку 4.3 представлена схема структурна компонентів, на якій відображені компоненти, що використовуються при наданні рекомендації страв в ресторані.

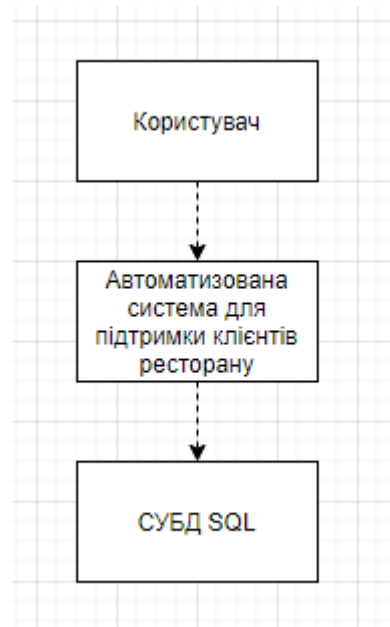


Рисунок 4.3 – Схема структурна компонентів

Висновки до розділу

В даному розділі було здійснено опис програмного та технічного забезпечення, а також архітектури програмного забезпечення. Наведено детальний опис засобів розробки, вимог до технічного забезпечення, а саме вимоги до клієнтських робочих станцій, серверу та серверу СУБД. У даному розділі наведені діаграми класів, послідовності та компонентів.

5. Технологічний розділ

5.1 Керівництво користувача

Для використання системи необхідно запустити застосунок, після чого відкриється головна форма програми.

Якщо клієнт вже відвідував ресторан раніше, то він може просто авторизуватись і отримати список рекомендацій (Рисунок **Error! No text of specified style in document..1**).

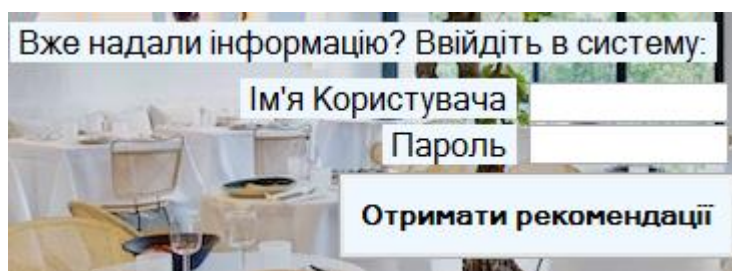


Рисунок **Error! No text of specified style in document..1** – Основна інформація про клієнта

Якщо ж клієнт вперше в закладі, то йому пропонується заповнити анкету з його перевагами (Рисунок 5.2).

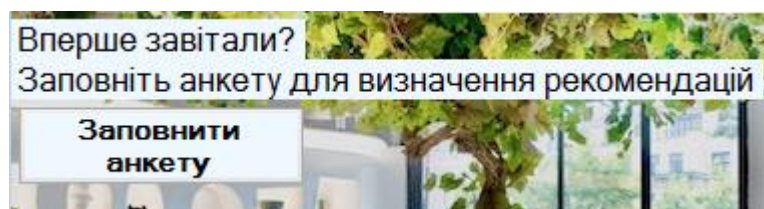


Рисунок 5.2 – Кнопка переходу в форму заповнення анкети

Після натискання на «Заповнити анкету», відкривається форма «Користувача». Вона зображена на Рисунку 5.3.

USER

MENU

Ім'я

Прізвище

Пароль

Повторити пароль

Вік

Цінова категорія

Переваги в виборі їжі

☐ М'ясна

☐ Рослинного походження

☐ Веганська

☐ Супи

☐ Салати

☐ Крупи

Напої

☐ Алкогольні

☐ Соки

☐ Газована вода

☐ Чай

Наявність алергій

Відправити анкету

Рисунок **Error! No text of specified style in document..3** – Перелік переваг на вибір користувача

Після надання відповідей на усі питання, необхідно натиснути на «Відправити анкету». Результатом цього буде надання рекомендації клієнту про вибір страви.

Після отримання рекомендації користувач може здійснити замовлення, а після кожного наступного приходу до ресторану просто авторизуватись в системі та знову отримати рекомендації, так як система вже має дані про нього.

5.2 Випробування програмного продукту

У цьому розділі описано тести та етапи їх впровадження, щоб перевірити, чи відповідає система функціональним вимогам, запропонованим

у технічному завданні системи для підтримки клієнтів ресторану.

5.2.1 Мета випробувань

Метою проведення випробувань є перевірка відповідності функцій системи для підтримки клієнтів ресторану та задоволення вимог технічного завдання.

5.2.2 Загальні положення

Випробування проводяться на основі наступних документів:

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

5.2.3 Результати випробувань

В процесі тестування була перевірена функціональність системи. У наступних таблицях наведений перелік випробувань основних функціональних можливостей (табл. 5.1 – 5.3).

Таблиця **Error! No text of specified style in document..2** – Введення основної інформації про клієнта і отримання рекомендацій

Мета тесту:	Перевірка функції «Введення основної інформації про клієнта»
Початковий стан КЗ	Відкрита «Головна» форма
Вхідні дані:	Поля «Головної» форми: ім'я, прізвище, пароль, вік.
Схема проведення тесту:	Ввести всю необхідну інформацію про клієнта та натиснути «Отримати

Мета тесту:	Перевірка функції «Введення основної інформації про клієнта»
	рекомендації»
Очікуваний результат:	Відкрита форма «Головна» з наданою рекомендацією в таблиці чи неспроможністю її надати
Стан системи після проведення випробувань:	Відкрита форма «Головна» з наданою рекомендацією в таблиці

Таблиця **Error! No text of specified style in document..3** – Введення інформації про переваги клієнта і занесення даних в базу.

Мета тесту:	Перевірка функції «Заповнення клієнтом анкети»
Початковий стан КЗ	Відкрита «Користувачка» форма
Вхідні дані:	Всі поля форми
Схема проведення тесту:	Ввести дані про зацікавлення та переваги клієнта при виборі страви, а також особисту інформацію для здійснення реєстрації в системі.
Очікуваний результат:	Занесення даних в базу, а також генерація рекомендацій для клієнта. Повернення до «Головної» форми.
Стан системи після проведення випробувань:	Відкрита «Головна» форма.

Таблиця **Error! No text of specified style in document..3** – Вхід користувача в форму для заповнення анкети

Мета тесту:	Перевірка функції входу в форму «Користувача»
Початковий стан КЗ	Відкрито «Головну» форму.
Вхідні дані:	Немає
Схема проведення тесту:	Натиснути кнопку «Заповнити анкету»
Очікуваний результат:	Відкрита сторінка «Користувач» для заповнення анкети чи неспроможність її відкрити
Стан системи після проведення випробувань:	Відкрита сторінка «Користувач»

Висновок до розділу

В даному розділі було наведено детальне керівництво користувача. Описано всі необхідні дії для виконання основних функцій автоматизованої системи. Наявність ілюстрацій робить керівництво більш зрозумілим. Були проведені випробування автоматизованої системи. Визначено їх мету, основні положення та результати.

ЗАГАЛЬНІ ВИСНОВКИ

Під час розробки дипломного проекту було поставлено задачу здійснення підтримки клієнтів ресторану шляхом надання рекомендацій щодо вибору страв. Результатом роботи є описане предметне середовище, функціональна модель, процес діяльності системи для підтримки клієнтів ресторану. Було також проведено аналіз наявних аналогів із зазначенням їхніх переваг та недоліків. За результатами аналізу результатів була обґрунтована необхідність створення такої системи, що могла б надати рекомендації щодо вибору меню, ґрунтуючись на персональних перевагах клієнта.

Інформаційне забезпечення системи включає опис вхідних та вихідних даних, а також структуру бази даних. Інтерфейс програми представлений за допомогою екранних форм. Робота системи передбачає надання рекомендацій щодо вибору страв відповідно до переваг клієнту, що він надав у вхідних даних запиту. Клієнт також може використати попередньо надані дані для авторизації в системі та швидкого отримання рекомендацій.

Була також визначена змістовна та математична постановка задачі. В якості можливих методів надання рекомендації розглядалися дерева прийняття рішень ID3, C4.5. Проведений аналіз наведених алгоритмів засвідчив необхідність розробки власного математичного методу. Тому в запропонованому алгоритмі побудови дерева рішень листи дерева рішень містять пару (рекомендовані страви та рівень довіри рекомендації), тобто нечітке дерево рішень

Було здійснено опис засобів розробки програмного забезпечення та архітектури, а також вимог до апаратного забезпечення. Також розроблено

					ДП 6331.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

керівництво користувача і проведені випробування над створеним програмним продуктом.

Перелік посилань

1. Алгоритм C4.5 [Електронний ресурс] // John Ross Quinlan. – 2012. – Режим доступу до ресурсу: <https://wiki.loginom.ru/articles/algorithm-C-4-5.html>.
2. Decision Trees – C4.5 [Електронний ресурс]. – 2013. – Режим доступу до ресурсу: <https://octaviansima.wordpress.com/2011/03/25/decision-trees-c4-5/>.
3. Quinlan J. R. C4.5: Programs for Machine Learning. — San Mateo: Morgan Kaufmann Publishers Inc., 1993. — 302 p.
4. Жернакова О. Системи рекомендацій і пошуку відеоконтента, 2012.
5. IV Всеукраїнська заочна науково-практична конференція «Сучасні інформаційні технології» Київ: 2013. – 107 с.
6. Ю. С. Нефедова, “Архитектура гибридной рекомендательной системы GEFEST (Generation–Expansion–Filtering–Sorting–Truncation)”, Системы и средства информ., 22:2 (2012), 176–196
7. Карл В. Разработка требований к программному обеспечению - М.: Русская Редакция, 2004. - 576 с.
8. R. Quinlan. Induction of decision trees. Machine Learning, 1986. - 81-106 p.
9. Witten, I. H. (Ian H.) Data mining : practical machine learning tools and techniques. – 3rd ed. / Ian H. Witten, Frank Eibe, Mark A. Hall. – Burlington, MA, USA, Morgan Kaufmann, 2011. – 664 p.
10. Ross Quinlan. C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo, CA, 1993.

11. Паклин Н.Б., Орешков В.И. Бизнес-аналитика: от данных к знаниям: Учебное пособие. — Питер, 2013. — С. 444-459.
12. Eibe Frank, Ian H. Witten: Generating Accurate Rule Sets Without Global Optimization. In: Fifteenth International Conference on Machine Learning, 1998 - 144-151 p.
13. J. Cendrowska. PRISM: An algorithm for inducing modular rules. International Journal of Man-Machine Studies, 1987. - 349-370.
14. Umano M., Okamoto H., Hatono I., Tamura H., Kawachi F., Umedzu S., Kinoshita J. Fuzzy decision trees by fuzzy ID3 algorithm and its application to diagnosis systems. - Fuzzy Systems, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the Third IEEE Conference on, 2113 - 2118 vol.3
15. Прохоренок Н. А. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера. -3-е изд., перераб. и доп. - СПб.: БХВ-Петербург, 2010. - 912 с.
16. Мерсер, Дэйв У., Кент, Аллан, Новицки, Стивен, Мерсер, Дэвид, Скуайер, Дэн, Чой, Ван Кью. М52 PHP 5 для начинающих. : Пер. с англ. М. : ООО И.Д. “Вильямс”, 2006. 848 с.
17. Котеров, Д. В. PHP 5 / Д. В. Котеров, А. Ф. Костарев. - 2-е изд., перераб. и доп. - СПб.: БХВ-Петербург, 2008. - 1104 с.

Додаток А***Тексти програмного коду***
Система для підтримки клієнтів ресторану

(Найменування програми (документа))

DVD-R

(Вид носія даних)

6 арк, 74 Кб

(Обсяг програми (документа) , арк.,) Кб)

Київ – 2020 року

Змн.	Арк.	№ докум.	Підпис	Дата

ДП 6311.00. 1081с ПЗ

Арк.

41

Текст програмного коду реалізації пошуку рекомендацій

```
int numUsers = 50;
```

```
int numItems = 10;
```

```
int numTraits = 2;
```

```
Variable<int> numObservations = Variable.Observed(100);
```

```
int numLevels = 2;
```

```
Range user = new Range(numUsers);
```

```
Range item = new Range(numItems);
```

```
Range trait = new Range(numTraits);
```

```
Range observation = new Range(numObservations);
```

```
Range level = new Range(numLevels);
```

```
using (Variable.ForEach(observation)) {
```

```
    VariableArray<double> products = Variable.Array<double>(trait);
```

```
    products[trait] = userTraits[userData[observation]][trait] *  
    itemTraits[itemData[observation]][trait];
```

```
    Variable<double> bias = (userBias[userData[observation]] +  
    itemBias[itemData[observation]]);
```

```
    Variable<double> affinity = (bias + Variable.Sum(products)); Variable<double>  
    noisyAffinity = Variable.GaussianFromMeanAndVariance(affinity,  
    affinityNoiseVariance);
```

```
    VariableArray<double> noisyThresholds = Variable.Array<double>(level);
```

```

        noisyThresholds[level] =
Variable.GaussianFromMeanAndVariance(userThresholds[userData[observation]]
[level], thresholdsNoiseVariance);
        ratingData[observation][level] = noisyAffinity > noisyThresholds[level];
    }

```

```

Variable.Array(Variable.Array<double>(numColumns), numRows)
Variable.Array(Variable.Array<double>(numRows), numColumns)
VariableArray<VariableArray<double>, double[][]>

```

```

var userData = Variable.Array<int>(observation);
var itemData = Variable.Array<int>(observation);
var ratingData = Variable.Array(Variable.Array<bool>(level), observation);

```

```

userTraits[user][trait] = Variable<double>.Random(userTraitsPrior[user][trait]);
itemTraits[item][trait] = Variable<double>.Random(itemTraitsPrior[item][trait]);
userBias[user] = Variable<double>.Random(userBiasPrior[user]);
itemBias[item] = Variable<double>.Random(itemBiasPrior[item]);
userThresholds[user][level] =
Variable<double>.Random(userThresholdsPrior[user][level]);

```

```

var userTraitsPrior = Variable.Array(Variable.Array<Gaussian>(trait), user);
var itemTraitsPrior = Variable.Array(Variable.Array<Gaussian>(trait), item);
var userBiasPrior = Variable.Array<Gaussian>(user);
var itemBiasPrior = Variable.Array<Gaussian>(item);

```

```
var userThresholdsPrior = Variable.Array(Variable.Array<Gaussian>(level), user);
```

```
var userTraits = Variable.Array(Variable.Array<double>(trait), user);
```

```
var itemTraits = Variable.Array(Variable.Array<double>(trait), item);
```

```
var userBias = Variable.Array<double>(user);
```

```
var itemBias = Variable.Array<double>(item);
```

```
var userThresholds = Variable.Array(Variable.Array<double>(level), user);
```

```
Gaussian traitPrior = Gaussian.FromMeanAndVariance(0.0, 1.0);
```

```
Gaussian biasPrior = Gaussian.FromMeanAndVariance(0.0, 1.0);
```

```
userTraitsPrior.ObservedValue      =      Util.ArrayInit(numUsers,      u      =>
Util.ArrayInit(numTraits, t => traitPrior));
```

```
itemTraitsPrior.ObservedValue      =      Util.ArrayInit(numItems,      i      =>
Util.ArrayInit(numTraits, t => traitPrior));
```

```
userBiasPrior.ObservedValue = Util.ArrayInit(numUsers, u => biasPrior);
```

```
itemBiasPrior.ObservedValue = Util.ArrayInit(numItems, i => biasPrior);
```

```
userThresholdsPrior.ObservedValue = Util.ArrayInit(numUsers, u =>
    Util.ArrayInit(numLevels,  l  =>  Gaussian.FromMeanAndVariance(1  -
numLevels / 2.0 + 0.5, 1.0)));
```

```
HashSet<int> visited = new HashSet<int>();
```

```
for (int observation = 0; observation < numObservations; observation++) {
```

```
    int user = Rand.Int(numUsers);
```

```
    int item = Rand.Int(numItems); int userItemPairID = user * numItems + item;
```

```
// pair encoding
```

```
    if      (visited.Contains(userItemPairID))      //      duplicate      generated      {
```

```

        observation--; // reject pair continue;
    }
    visited.Add(userItemPairID);
    double[] products = Util.ArrayInit(numTraits, t => userTraits[user][t] *
itemTraits[item][t]);
    double bias = userBias[user] + itemBias[item];
    double affinity = bias + products.Sum();
    double noisyAffinity = new Gaussian(affinity, affinityNoiseVariance).Sample();
    double[] noisyThresholds = Util.ArrayInit(numLevels, 1 => new
Gaussian(userThresholds[user][l], thresholdsNoiseVariance).Sample());

    generatedUserData[observation] = user;
    generatedItemData[observation] = item;
    generatedRatingData[observation] = Util.ArrayInit(numLevels, 1 =>
noisyAffinity > noisyThresholds[l]);
}

```

```

Rand.Restart(12347);
double[][] userTraits = Util.ArrayInit(numUsers, u => Util.ArrayInit(numTraits, t
=> userTraitsPrior[u][t].Sample()));
double[][] itemTraits = Util.ArrayInit(numItems, i => Util.ArrayInit(numTraits, t
=> itemTraitsPrior[i][t].Sample()));
double[] userBias = Util.ArrayInit(numUsers, u => userBiasPrior[u].Sample());
double[] itemBias = Util.ArrayInit(numItems, i => itemBiasPrior[i].Sample());

```

```
double[][] userThresholds = Util.ArrayInit(numUsers, u =>
Util.ArrayInit(numLevels, l => userThresholdsPrior[u][l].Sample()));
```

```
for (int i = 0; i < numTraits; i++) { // Assume that numTraits < numItems
    for (int j = 0; j < numTraits; j++) {
        itemTraitsPrior.ObservedValue[i][j] = Gaussian.PointMass(0);
    }
    itemTraitsPrior.ObservedValue[i][i] = Gaussian.PointMass(1);
}
```

```
numObservations.ObservedValue = 1;
userData.ObservedValue = new int[] { 5 };
itemData.ObservedValue = new int[] { 6 };
ratingData.ClearObservedValue();
```

```
Bernoulli[] predictedRating = engine.Infer<Bernoulli[][]>(ratingData)[0];
Console.WriteLine("Predicted rating:");
foreach (var rating in predictedRating) Console.WriteLine(rating);
```

```
var userTraitsPosterior = engine.Infer<Gaussian[][]>(userTraits);
var itemTraitsPosterior = engine.Infer<Gaussian[][]>(itemTraits);
var userBiasPosterior = engine.Infer<Gaussian[]>(userBias);
var itemBiasPosterior = engine.Infer<Gaussian[]>(itemBias);
var userThresholdsPosterior = engine.Infer<Gaussian[][]>(userThresholds);
```

```
userTraitsPrior.ObservedValue = userTraitsPosterior;  
itemTraitsPrior.ObservedValue = itemTraitsPosterior;  
userBiasPrior.ObservedValue = userBiasPosterior;  
itemBiasPrior.ObservedValue = itemBiasPosterior;  
userThresholdsPrior.ObservedValue = userThresholdsPosterior;
```

```
userTraitsPrior.ObservedValue = engine.Infer<Gaussian[][]>(userTraits);  
itemTraitsPrior.ObservedValue = engine.Infer<Gaussian[][]>(itemTraits);
```

```
InferenceEngine engine = new InferenceEngine();  
engine.Compiler.GivePriorityTo(typeof(GaussianProductOp_SHG09))
```

ДП 6331.00.000 ПЗ

					ДП 6331.00.000 ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО”
Кафедра автоматизованих систем обробки інформації та управління

УЗГОДЖЕНО

Керівник проєкту

_____ Олена ЖДАНОВА
(підпис) (вл. ім'я, прізвище)

“13” квітня 2020 р.

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ Олександр ПАВЛОВ
(підпис) (вл. ім'я, прізвище)

“14” квітня 2020 р.

Система з підтримки роботи з клієнтами ресторану

ТЕХНІЧНЕ ЗАВДАННЯ

Шифр *ДП 6331.01.000 ТЗ*

на 8 сторінках

Київ – 2020 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО”
Кафедра автоматизованих систем обробки інформації та управління

УЗГОДЖЕНО

Керівник проєкту

_____ Олена ЖДАНОВА
(підпис) (вл. ім'я, прізвище)

“13” квітня 2020 р.

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ Олександр ПАВЛОВ
(підпис) (вл. ім'я, прізвище)

“14” квітня 2020 р.

Система з підтримки роботи з клієнтами ресторану

ТЕХНІЧНЕ ЗАВДАННЯ

Шифр *ДП 6331.01.000 ТЗ*

на 10 сторінках

Київ – 2020 року

ДП 6331.00.000 ТЗ

ДП 6331.00.000 ТЗ

					ДП 6331.00.000 ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

Зміст

<u>1</u>	<u>ЗАГАЛЬНІ ПОЛОЖЕННЯ</u>	11
<u>1.1</u>	<u>Повне найменування системи</u>	11
<u>1.2</u>	<u>Найменування організації-замовника та організацій-учасників робіт</u>	11
<u>1.3</u>	<u>Перелік документів, на підставі яких створюється система</u>	11
<u>1.4</u>	<u>Планові терміни початку і закінчення роботи зі створення системи</u>	11
<u>2</u>	<u>ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ</u>	12
<u>2.1</u>	<u>Призначення системи</u>	12
<u>2.2</u>	<u>Цілі створення системи</u>	12
<u>3</u>	<u>ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ</u>	13
<u>4</u>	<u>ВИМОГИ ДО ПРОГРАМИ</u>	16
<u>4.1</u>	<u>Вимоги до функціональних характеристик</u>	16
<u>4.2</u>	<u>Вимоги до надійності</u>	16
<u>4.3</u>	<u>Вимоги до складу і параметрів технічних засобів</u>	16
<u>5</u>	<u>СТАДІЇ І ЕТАПИ РОЗРОБКИ</u>	17
<u>6</u>	<u>ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ</u>	18
<u>6.1</u>	<u>Об'єм випробувань</u>	18

ДП 6331.01.000 ТЗ

					ДП 6331.01.000 ТЗ						
Зм.	Арк.	Прізвище	Підпис								
Розроб.		Яворський А.С.			Система з підтримки роботи з клієнтами ресторану			Літ.	Лист	Листів	
Перевірів.		Жданова О.Г.		Дата						2	17
								КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-63			
Н. кон.		Проскура С.Л.									
Затв.		Павлов О.А.									

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Повне найменування системи

Система з підтримки роботи з клієнтами ресторану.

1.2 Найменування організації-замовника та організацій-учасників робіт

Замовником системи є кафедра автоматизованих систем обробки інформації і управління факультету інформатики та обчислювальної техніки НТУУ «КПІ». Розробником системи є Яворський Антон Сергійович.

1.3 Перелік документів, на підставі яких створюється система

Підставою для розробки системи є завдання на переддипломну практику.

1.4 Планові терміни початку і закінчення роботи зі створення системи

Плановий термін початку робіт по створенню системи – 01.05.2020.
Плановий термін завершення робіт по створенню системи – 30.05.2020.

					ДП 6331.00.000 ТЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ

2.1 Призначення системи

Система призначена для підтримки взаємодії між закладом харчування та власне клієнтом, та надання рекомендацій щодо меню замовлень згідно побажань клієнтів ресторану.

2.2 Цілі створення системи

Метою є мінімізація зусиль для оформлення замовлення шляхом використання автоматично створених рекомендацій.

Для досягнення поставленої мети необхідно реалізувати такі задачі:

- а) ведення страв;
- б) ведення типів замовлень;
- в) ведення питань щодо побажань клієнтів;
- г) формування рекомендацій;
- д) формування звітності(меню як звітність).

					ДП 6331.00.000 ТЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

3 ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ

Об'єктом автоматизації є процес створення типів замовлення та рекомендацій для кожного клієнта, а також формування звітності щодо виконання цих задач. Надамо декілька специфічних для ресторанного бізнесу термінів для кращого розуміння проблеми:

– **Аванзал** - приміщення в ресторані для очікування, збору та відпочинку учасників свята.

– **Алкогільні напої** - горілка, настоянки, лікєро-горілчані вироби, виноградні та плодово- ягідні вина, коньяки.

– **Асортиментний мінімум** - певна кількість страв та напоїв, яка повинна бути у продажу кожного дня.

– **Банкет** - святковий званий обід або вечеря, який організовують на честь кого-небудь або чого-небудь.

– **Банкетний зал** - основне приміщення ресторану, де відвідувачі обідають або вечеряють, відпочивають або справляють ювілей (свято).

– **Банкет з повним обслуговуванням** - форма обслуговування банкету, коли всі учасники свята сидять за сервірованим столом, на якому не ставлять закуски, страви, напої, а їх подають офіціанти в обнос.

– **Бізнес-ланч** - обід для ділових людей у будні з 12 до 16 години за окремим меню, яке містить 4-5 холодних закусок, 2-3 перші страви, 3-4 другі страви, 2-3 десерти та гарячі напої (чай, каву).

– **Бренд** - відома торгова марка.

					ДП 6331.00.000 ТЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

– **Дегустація** - куштування їжі, напою; зняття проби з вина.

– **Делікатес** - вишукана, тонка, дорога страва з м'ясних, рибних, фруктових продуктів, яку зазвичай подають під час святкових званих обідів, банкетів, прийомів.

– **Десерт** - солодкі страви або фрукти, які подають наприкінці обіду, вечері. Десертні страви зазвичай включаються до меню, за яким обслуговуються індивідуальні туристи та групи туристів.

– **Закуска** - характер і форма страви, яку подають самостійно або у складі обіду; може бути холодною та гарячою.

– **Закупний товар** - товар, що його купує заклад ресторанного господарства для подальшого перепродажу споживачам без оброблення у закладі.

– **Замовлення** - письмове або усне доручення будь-кому щонебудь виготовити, зробити, продати, наприклад, замовити їжу в ресторані, квитки до театру тощо.

– **Коктейль** - суміш алкогольних і безалкогольних напоїв із додаванням цукру, прянощів, фруктів і деяких інших компонентів. Зазвичай коктейль включається до прохолодних напоїв, які пропонують відвідувачам ресторанів, кафе, барів.

– **Кухня** - стиль приготування і асортимент ресторанних страв, наприклад, європейська, східна, китайська кухня; приміщення для приготування їжі або штат кухарів.

– **Ланч** - час надання - від 10.00 до 14.00. Прийнято проводити у святкові дні.

					ДП 6331.00.000 ТЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

– **Страва** - одна зі змін страв, які подають під час сніданку, обіду або вечері, наприклад, закуска, бульйон (суп), основна гаряча страва, десерт

– **Страва дня** - головна страва в меню, яка пропонується відвідувачам ресторану тільки у конкретний день. Зазвичай про таку страву повідомляється на початку меню під девізом «шеф-повар рекомендує».

– **Торговельний зал ресторану** - основне приміщення, де обслуговують відвідувачів.

Далі наведений опис процесів і розподіл відповідальності між спеціалістами.

Коли клієнт вперше заходить до ресторану, працівник ресторану надає йому анкету або усно дізнається про його переваги. Після цього дані заносяться в систему, що власне й надає рекомендації щодо вибору страви.

Адміністратор системи здійснює періодичні оновлення та виправлення неточностей даних, занесених в систему.

Менеджер ресторану слідкує за виконанням всіх процесів, відповідає та регулює роботу між персоналом ресторану та системою.

					ДП 6331.00.000 ТЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

4 ВИМОГИ ДО ПРОГРАМИ

4.1 Вимоги до функціональних характеристик

Система має реалізовувати наступні функції:

- функція ведення страв;
- функція ведення типів замовлення;
- функція ведення питань щодо побажання клієнтів
- функція формування рекомендацій;
- функція формування звітності.

4.2 Вимоги до надійності

Спеціальних вимог щодо надійності системи немає.

4.3 Вимоги до складу і параметрів технічних засобів

Технічні засоби, що використовуються під час проведення випробувань: IBM-сумісний комп'ютер, що включає:

- процесор Pentium II 400 або кращий;
- оперативна пам'ять – 128 мб або більше;
- жорсткий диск – 20 мб вільного місця або більше.

Програмні засоби, що використовуються під час проведення випробувань – операційна система Windows 7 або вище.

Випробування мають проводитися в нормальних кліматичних умовах по ГОСТ 22261-94.

					ДП 6331.00.000 ТЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

5 СТАДІЇ І ЕТАПИ РОЗРОБКИ

Вивчення рекомендованої літератури; аналіз існуючих розробок; постановка задачі; розробка інформаційного, програмного та математичного забезпечення; оформлення пояснювальної записки.

					ДП 6331.00.000 ТЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

Випробування проводяться на основі наступних документів:

–ГОСТ 34.603-92. Інформаційна технологія. Види випробувань автоматизованих систем;

–ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

Прийомні випробування мають проводитися на комп'ютері замовника в строки не пізніше 15.05.2020. Прийомні роботи проводяться згідно створеної виконавцем до того часу Програми і методики випробувань, що має бути узгоджена з замовником.

Перелік документів, що мають бути пред'явлені на випробування

- технічне завдання;
- програма і методика випробувань;
- керівництво користувача.

6.1 Об'єм випробувань

Етапи випробувань:

- 1) ознайомчий;
- 2) випробування.

На ознайомчому етапі проводиться:

- 1) перевірка комплектності програмної документації;
- 2) перевірка комплектності складу технічних і програмних засобів.

Під час етапу випробувань проводиться:

- 1) перевірка відповідності технічних характеристик системи;
- 2) перевірка ступеню виконання вимог функціонального призначення системи.

					ДП 6331.00.000 ТЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

Власник документу:

Попенко Володимир Дмитрович

Дата перевірки:

10.06.2020 02:24:53 EEST

Дата звіту:

10.06.2020 14:55:23 EEST

ID перевірки:

1003919979

Тип перевірки:

Doc vs Internet + Library

ID користувача:

77149

Назва документу: Javorskij_bachelor_is63_2

ID файлу: 1003935287 Кількість сторінок: 33

Кількість слів: 4670 Кількість символів: 34229 Розмір файлу: 2.04 MB

8.57% Схожість

Найбільша схожість: 4.13% з джерело бібліотеки. ID файлу: 1003798194



5.27% Схожість з Інтернет джерелами

..... Page 35



7.54% Текстові збіги по Бібліотеці акаунту

18 Page 36

0% Цитат



Не знайдено жодних цитат

0% Вилучень



Вилучений текст відсутній

Підміна символів



Не знайдено замінених символів

Графічний матеріал до дипломного проєкту

на тему: Система з підтримки роботи з клієнтами ресторану

Київ – 2020 року



ДП ІС-6331.02.000 ССВ

					Схема структурна варіантів використання			
Зм.	Арк	№ документа	Підпис	Дата				
Розробив		Яворський А.С.			Система для підтримки клієнтів ресторану			
Перевішив		Жданова О.Г.						
Т. кон.					КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-63			
Н. кон.								
Затвердив		Жданова О.Г.						

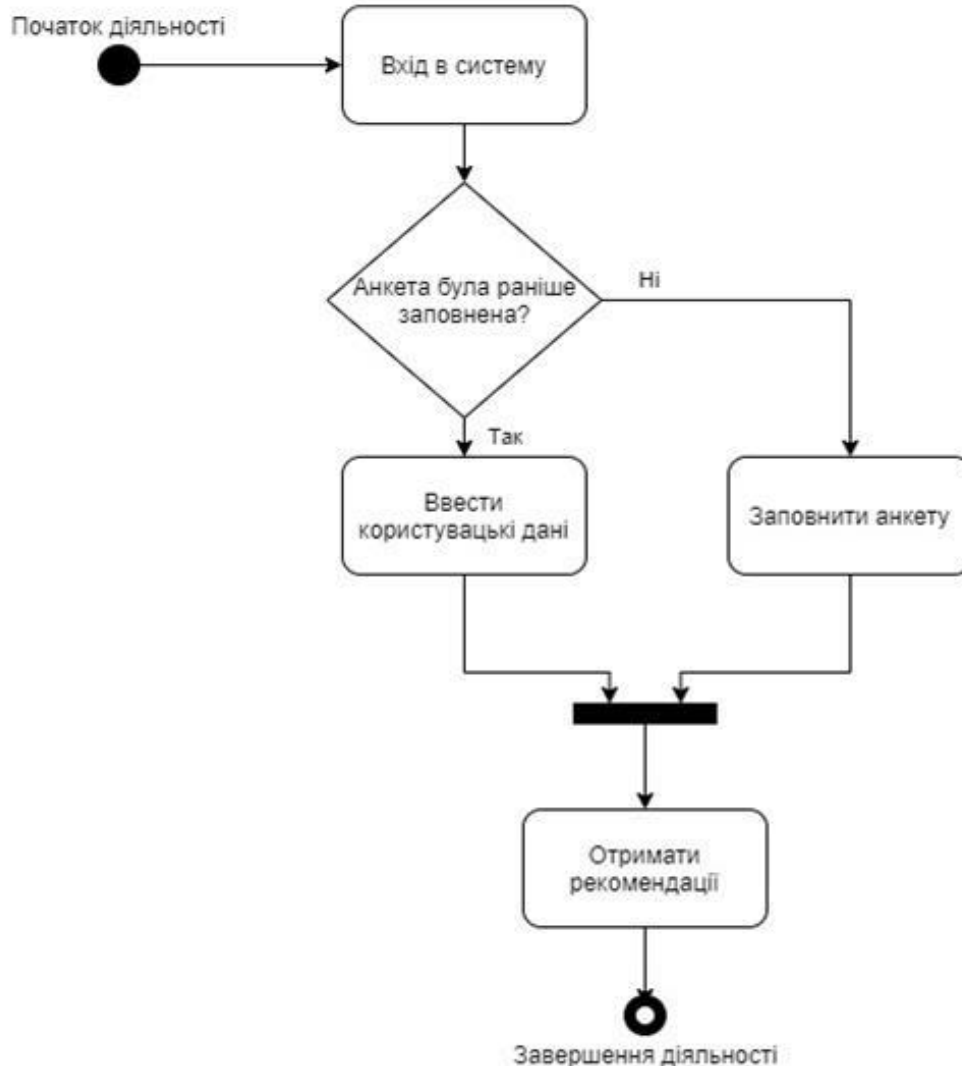
Літера

Маса

Масштаб

Аркуш 1

Аркушів 1



					ДП ІС-6331.03.000 ССС				
					Схема структурних станів системи	Літера		Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата					
Розробив	Яворський А.С.								
Перевірив	Жданова О.Г.					Аркуш 1		Аркушів 1	
Т. кон.					Система для підтримки клієнтів ресторану	КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-63			
Н. кон.									
Затвердив	Жданова О.Г.								

Страви

Id страви	Integer
Назва страви	String
Складові	String

Замовлення

Id замовлення	Integer
Номер страви	String
Назва страви	String
Загальна вартість замовлення	Integer

Меню

Назва меню	Integer
Id страви	String
Назва страви	String
Складові страви	String
Ціна	Integer

Рекомендація

Id рекомендації	Integer
Номер рекомендованої страви	String
Назва рекомендованої страви	String
Короткий опис рекомендації	String

Анкета

Id клієнта	Integer
Користувачське ім'я клієнта	String
Пароль	String
Наявність алергій	Boolean
Складові страв	String array
Цінова категорія	Integer
Вид меню	String

ДП ІС-6331.04.000 СБД

Схема бази даних

Літера

Маса

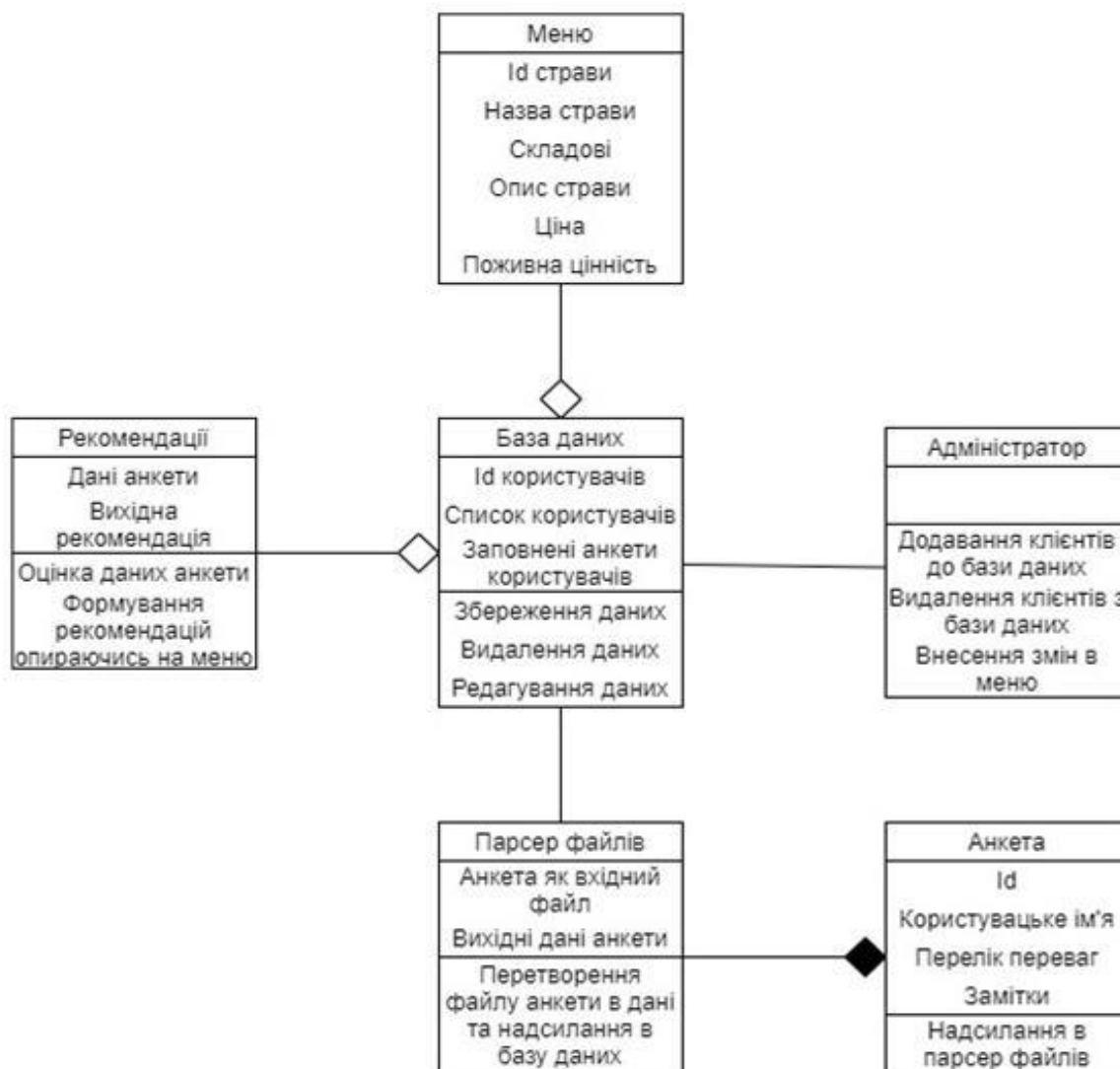
Масштаб

Аркуш 1

Аркушів 1

Система для підтримки клієнтів
ресторануКПІ ім. Ігоря Сікорського
кафедра АСОІУ гр. ІС-63

Зм.	Арк.	№ документа	Підпис	Дата
Розробив		Яворський А.С.		
Перевірив		Жданова О.Г.		
Т. кон.				
Н. кон.				
Затвердив		Жданова О.Г.		



					ДП ІС-6331.05.000 ССК				
					Схема структурних класів програмного забезпечення	Літера		Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата					
Розробив		Яворський А.С.							
Перевірив		Жданова О.Г.				Аркуш 1		Аркушів 1	
Т. кон.					Система для підтримки клієнтів ресторану	КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-63			
Н. кон.									
Затвердив		Жданова О.Г.							

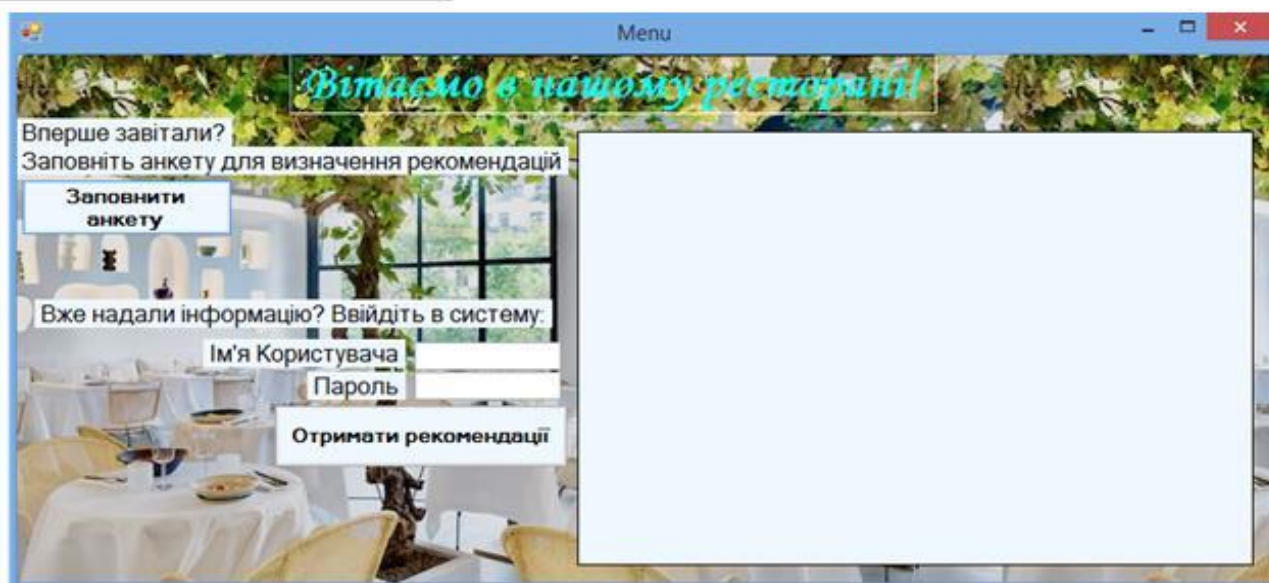


Рисунок 1 – Екранна форма «Головного меню»



Рисунок 2 – Екранна форма «Користувача»

					ДП ІС-6331.06.000 КЕ			
Зм.	Арк.	№ документа	Підпис	Дата	Креслення вигляду екранних форм	Літера	Маса	Масштаб
Розробив		Яворський А.С.						
Перевірив		Жданова О.Г.			Система для підтримки клієнтів ресторану	Аркуш 1		Аркушів 1
Т. кон.		Прескура С.Л.				КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-63		
Н. кон.								
Затвердив		Павлов О.А.						